Prototyping a Digital Postbox

BSc Applied Computing Final Year Project

Written by: Amy Burgess
Supervisor: Andrew Dearden

2009-05-01
Revision 105

Table of Contents

1 Introduction 4

1.1 Abstract 4

1.2 Basic Requirements 4

2 Technology Review 4

2.1 Hardware 5

2.1.1 OpenWRT 5

2.1.2 DD-WRT 7

2.1.3 Mini-ITX 7

2.1.4 Peripherals 8

2.1.4.1 USB Storage 8

2.1.4.2 Modems 8

2.1.4.3 Bluetooth Adapters 8

2.1.4.4 Landline Connections 9

2.1.5 Mobile Handsets 10

2.1.5.1 Apple iPhone 10

2.1.5.2 Google Android 10

2.2 Software 11

2.2.1 Symbian 11

2.2.2 Linux 12

2.3 Summary 12

3 Development 12

3.1 Hardware Installation 13

3.2 Software Installation 15

3.3 Mobile Software 15

3.4 Postbox Software 16

4 Testing 17

5 Critical Evaluation 18

6 References 18

6.1 Credits and Thanks....18

6.2 Bibliography....18

7 Appendixes....19

7.1 Appendix 1: OpenWRT Analysis....19

7.2 Appendix 2: Python Code....19

7.3 Appendix 3: List of Figures....19

1 Introduction

1.1 Abstract

This document examines the problems with limited bandwidth in remote areas of developing countries and some of the potential solutions to the lack of bandwidth in those areas. Also discussed are the relevant potential hardware and software technologies required to implement a digital postbox.

An exploration of various software and hardware configurations charts the evolution of the project from concept to functional prototype. A brief discussion of the various testing methods employed in such a complex system concludes the main report.

1.2 Basic Requirements

In developing countries access to good bandwidth for data or voice communications is severely limited, especially in rural areas. In the short-term, building high speed data infrastructure out to these areas is prohibitively expensive while the market for such lines is still immature.

Rather than take the infrastructure to the people I plan to take advantage of existing infrastructure and a phenomenon described by Chambers as “development tourism”, where people are willing to travel to places with better communication facilities in order to access various services.

To exploit this, I plan to create a device I am calling a digital postbox (DPB) located at the nearest village with access to a fixed phone line and allow the local people to remotely access the DPB to send and receive emails. Remote access will be gained via portable handheld devices, preferably, mobile phones.

In order to prove the concept of a digital postbox is valid and workable, the system should meet the following requirements:

- The mobile device must have a built-in digital camera
- The mobile device must have a wireless communication protocol similar to Bluetooth
- The DPB must be low powered
- The DPB must be able to process around 20 messages a week
- The DPB must have access to the internet
- The DPB must be as autonomous as possible with no input devices other than the mobile phone and no output devices other than the connection to the internet
- All parts of the system must be as cheap to source as possible

2 Technology Review

The project to build a digital postbox requires various pieces of hardware and accompanying software, in addition to the custom software needed to complete the system. The hardware must be extremely rugged, consume very little power yet be powerful enough to run the required software at a decent speed.

The basic concept is that a rural user will package up the documents into a compressed archive using the software provided on their mobile phone. Then as the user approaches the Bluetooth range of the digital postbox the software will automatically connect to it and any packages ready to be sent will be transferred and stored on the bridging device. When the landline becomes available the bridging device will connect to the routing server via the internet and transmit all the waiting outgoing packages. Once all the packages have been successfully transferred the bridging device will delete the messages and disconnect. The server will then email the packages on to the correct recipient. The concept diagram below illustrates the basic principle.

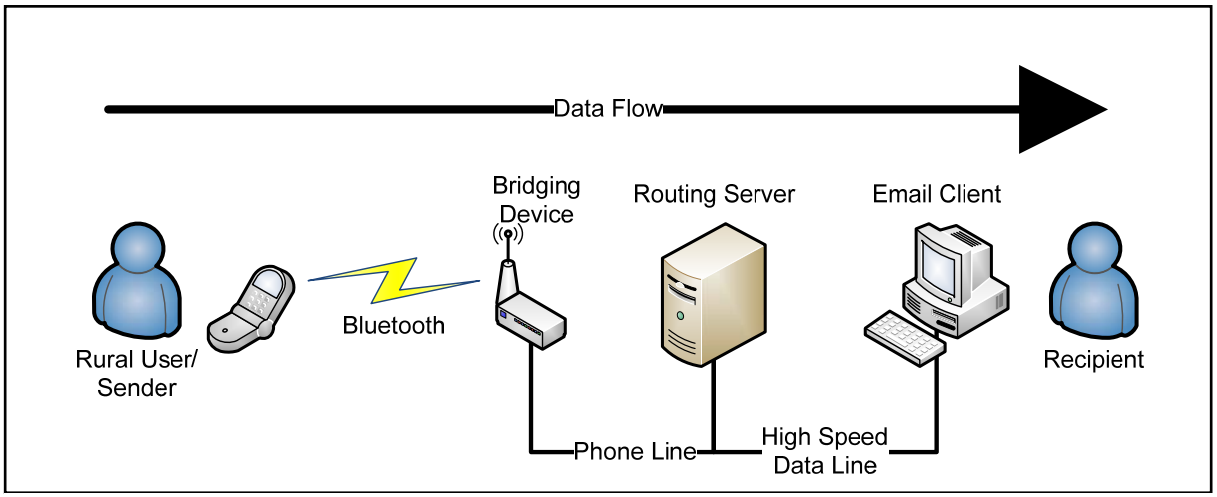

FIGURE 1: DIGITAL POSTBOX CONCEPT DIAGRAM

2.1 Hardware

Due to the harsh nature of the environments that the "field" hardware will have to endure during their operational lifetime it is essential that they are rugged, energy efficient and have a high tolerance to dust and heat. To this end equipment which is fitted with cooling fans or hard disk drives will be considered only after all other options have been discounted. Anything with moving parts is a concern for future maintenance. Moving parts generate friction which in turn generates heat. As the probable deployment areas are generally a lot warmer than the UK, introduction of additional sources of heat in the system should be avoided where possible.

Alongside the environmental requirements there are some basic hardware restrictions. These include:

- At least one USB port to enable Bluetooth connectivity
- Enough RAM to run the bridging software as well as the device's operating system
- Enough storage capacity to hold the operating system with enough spare capacity to store at least five 5MB message packages

2.1.1 OpenWRT

OpenWRT is a hardware platform that uses a stripped down Linux distribution specifically designed to run on wireless routers. The OpenWRT project is open-sourced under the GNU General Public License. One of the attractive things about this platform is that is designed to run on low powered devices that have no moving parts.

The OpenWRT website has an extensive list of hardware[1] that supports OpenWRT, over 400 devices. The site lists the minimum requirements to run OpenWRT as 4MB of flash storage and 16MB of RAM[2], and of course our requirement of a USB port. These specifications are numeric making it very easy to do analysis on the hardware list to find the most suitable device, or devices, for the successful execution of the project.

Once I had the hardware list in Excel I had a lengthy task of tidying up the data, a lot of the hardware list had missing information. A long time was spent on various router manufacturer websites locating the missing data. Even then there were several devices for which I could not locate any specifications for.

The hardware list has a field called "Status" which refers to the level of software compatibility. To ease the project implementation I decided to only use devices that are confirmed to run OpenWRT successfully. The complete list of statuses is as follows:

- Kamikaze - Will run the latest Kamikaze Linux distribution
- Supported - Will run the previous White Russian distribution
- Untested - Should work, not officially tested
- WIP - Work In Progress

This meant only devices flagged with either Kamikaze or Supported would be considered for the project.

The basic level of the analysis consisted of creating a lookup that checked the RAM and flash capacity were at least equal to or exceeded the minimum software requirements. Additionally the lookup checked if the device had USB. To further breakdown the analysis I coloured the specifications as follows:

- Red – Fails to meet the minimum requirements
- Orange – Meets the minimum requirements
- Green – Exceeds the minimum requirements

This colour coding makes the analysis much easier to read and at a glance you can tell which devices would be the most suitable. Of the 425 devices listed only 19 met the minimum requirements as previously outlined.

Manufacturer	Model	Version	Flas	RA	USB	USB_Por	Status	Suitabili	Cost	Notes
4gSystems	AccessCube	MTX-1	32	64	Yes	1	Kamikaze	Yes		Obsolete
Asus	WL-500b	1	4	16	Yes	1	Supported	Yes		Obsolete
Asus	WL-500g		4	16	Yes	1	Supported	Yes		Obsolete
Asus	WL-500g Deluxe		4	32	Yes	2	Supported	Yes		Obsolete
Asus	WL-500g Premium	1	8	32	Yes	2	Supported	Yes	£73.28	End Of Line
Asus	WL-HDD2.5		4	16	Yes	1	Supported	Yes		Obsolete & HDD
Belkin	F5D7231-4P	1001ef	4	16	Yes	1	Supported	Yes		Obsolete
Compex	NP18A	Retail	4	32	Yes	4	Kamikaze	Yes		OEM Customers Only
Compex	NP18A	OEM - IXP422	4	32	Yes	4	Kamikaze	Yes		OEM Customers Only
Compex	NP18A	OEM - IXP425	8	64	Yes	4	Kamikaze	Yes		OEM Customers Only
Dell	TrueMobile 2300		4	16	Yes	0	Supported	Yes		Built-in Bluetooth/Obsolete
Fon	La Fonera 2.0	FON2202	8	32	Yes	1	Kamikaze	Yes	£39.95	
Freecom	FSG-3		4	64	Yes	4	Kamikaze	Yes		Obsolete
Linksys	NSLU2		8	32	Yes	2	Supported	Yes		Obsolete
Linksys	WRTSL54GS	1.0	8	32	Yes	1	Supported	Yes		Obsolete
Linksys	WRTSL54GS	1.1	8	32	Yes	1	Supported	Yes		Obsolete
Mitsubishi	Mitsubshi (Diamond Data)	R100	4	16	Yes	1	Supported	Yes		Obsolete
Netgear	WGT634U	1	8	32	Yes	1	Kamikaze	Yes		Obsolete
Yakumo	Yakumo Wireless Storage	60	4	16	Yes	1	Supported	Yes		Obsolete

FIGURE 2: LIST OF SUITABLE OPENWRT COMPATIBLE HARDWARE

[1] CompleteTableOfHardware - http://wiki.openwrt.org/CompleteTableOfHardware (Accessed 2008-12-12)

[2] MinimumSystemRequirements - http://wiki.openwrt.org/MinimumSystemRequirements (Accessed 2008-12-12)

As you can see in the table above, out of those 19, four devices exceeded the minimum requirements. However the majority of the products listed are now considered obsolete by their manufacturers so sourcing one of those devices might be an issue.

2.1.2 DD-WRT

DD-WRT is similar to OpenWRT in the respect that it uses routers as a hardware platform making it ideal as the base for the bridging software. It differs to OpenWRT in the fact that it appears to be slightly better supported with regards to the available software. However there are potentially legal issues with using a dd-WRT system, there are various blogs that state the original author is now charging for licenses whilst still claiming to be free open-source software[3]. While blogs are not a valid source of information there are enough of them to cast doubt upon the viability of using dd-WRT.

2.1.3 Mini-ITX

Mini-ITX is a hardware format invent by the chip manufacturer VIA. It is based around an ultra small form factor motherboard. The board is 17cm by 17cm in size, allowing for it to be fitted into very small, compact cases[4].

The advantages of using a mini-ITX device in the project is that it meets the requirements of being low powered and can be kept cooled using passive, fan-less methods. Additionally mini-ITX provides access to processing power similar to that of a desktop computer; this is very attractive as it means the bridging software can be written as any normal software would without having to take special operational limitations into consideration.

Using a technology based around the X86 standard also means that a full-sized operating system can be used. Full operating systems are usually better supported so any problems that occur during installation will have solutions.

A mini-ITX solution gives immense flexibility with hardware setups as well as the ability to use standard, widely available PC components in its construction. One of the main disadvantages of mini-ITX is the higher cost of purchasing the hardware.

	FX5621		M61G		M73G	
	Stats	Cost	Stats	Cost	Stats	Cost
Case		£ 285.00		£ 129.00		£ 145.00
CPU Speed (GHz)	1		1		1	
RAM	512		512	£ 29.00	512	£ 13.50
USB Ports	2		6		6	
Storage (GB)	1	£ 19.00	1	£ 39.00	1	£ 39.00
	Total:	£ 304.00	**Total:**	£ 197.00	**Total:**	£ 197.50

FIGURE 3: A COMPARISON OF FAN-LESS MINI-ITX SYSTEMS

[3] DD-WRT - An affront to the good will of the F/OSS community http://www.bitsum.com/about-ddwrt.htm (Accessed 2009-01-12)

[4] Mini-itx.com – faq http://www.mini-itx.com/faq.asp#Hardware0 (Accessed 2009-01-05)

2.1.4 Peripherals

While the bridging device will be a major component in the system to be fully functional it will require various pieces of additional hardware to be connected to it.

2.1.4.1 USB Storage

To ensure that there is sufficient storage capacity to store up to a minimum of 25MB of messages in case of a temporary communications error it might be necessary to purchase a USB mass storage device (also known as a pendrive) and integrate it into the system.

These devices are able to store up to 64GB for a relatively low cost. 2GB versions are available for under £5 from most IT equipment retailers, this gives room for the operating system, the bridging software and the message overhead with plenty of room to spare. A pendrive with a capacity of 1GB or more could be used in conjunction with a mini-ITX system in lieu of using a traditional hard disk drive.

Another option would be to use an SD memory card similar to those found in digital cameras and mobile phones. They are available in capacities similar to pendrives. The downside to this method is that card readers aren't as widespread as USB pendrives so a card reader would need to be purchased alongside the card. Card readers are usually USB powered so it would be cheaper to source an USB pendrive in the first place.

2.1.4.2 Modems

Regardless of final device chosen as the bridge if it has to connect to an analogue line a 56k modem will have to be used. The interface to the bridge has to be via USB and the modem should be able to draw its power from solely the USB. This is based on the assumption that power outlets will be scarce.

Most modern hardware is equipped with at least one USB port. Serial ports are now considered obsolete and a modem connecting via a serial port would require its own power source.

While analogue modems are considered legacy products there are still several available, these are listed in the table below:

Manufacturer	Product Number	Estimated Cost	URL
US Robotics	USR805637	£26.80	http://www.usr.com/products/modem/modem-product.asp?sku=USR805637
Trust	MD-1270	£23.64	http://www.trust.com/products/product_detail.aspx?item=15529

FIGURE 4: A LIST OF USB POWERED ANALOGUE MODEMS

DSL modems should be provided by the ISP, either for free or for a small additional cost alongside the subscription, and can plug directly into any free Ethernet port on the postbox.

2.1.4.3 Bluetooth Adapters

Interfacing the mobile phone with the digital postboxes will be done using Bluetooth. This is preferable to using a USB connection because many phones use propriety connectors at the phone end. It would be logistically difficult to ensure that every possible connection would be catered for.

At the recent GSM World conference in Barcelona GSMA and about 17 other major mobile phone manufacturers announced that by 2012 there will be a universal standard charger based around a micro-USB connection[5]. While this means that only a single type of cable will be required to connect to all phones there is still the remaining problem of ensuring the phone's file system is mounted in the correct place. This becomes increasingly problematic if multiple devices are connected at the same time.

Connecting a phone via USB would mean having to mount the phone as a removable storage device. Once mounted the software would have to search for any packages that require sending, copy the package to its internal storage, delete the source file and cleanly dismount the phone. This is an overly complex process.

Using the phone to "push" the required packages via Bluetooth is a cleaner, easier and preferred method of data transfer in this system.

There is also the issue if USB cables are used that they may become detached from the postbox during everyday use resulting in the system failing to perform correctly.

Transfer rates for Bluetooth are about 700Kbps with a range of about 10 metres. Newer versions of the Bluetooth standard have ranges up to 100 metres[6]. This means a 5MB file will take roughly a minute to transfer to the postbox under normal conditions[7]. A short experiment using the chosen adapter at a later date will more accurately map the speed and potential range of the device.

USB powered Bluetooth adapters can be purchased from most major IT equipment suppliers for under £10.

2.1.4.4 Landline Connections

The landline component is a critical component to the system, it is also the component I will have the least say in choosing as it fully depends on the infrastructure in the target deployment area. The best possible connection will be a DSL line and in the worst case a 56K modem can be used.

Connection Speed	File Size (MB)	Transfer Time		
		Hours	Minutes	Seconds
56Kbps Modem	25	1	2	24
128Kbps ISDN	25	0	27	18
640Kbps DSL	25	0	5	27

FIGURE 5: MINIMUM THEORETICAL TRANSFER SPEEDS VIA A LANDLINE.
SOURCE: ICALC – BANDWIDTH CALCULATOR - HTTP://WWW.IBEAST.COM/CONTENT/TOOLS/BAND-CALC.ASP (ACCESSED 2008-12-12)

[5] Mobile Industry Unites to Drive Universal Charging Solution for Mobile Phones - http://www.gsmworld.com/newsroom/press-releases/2009/2548.htm (Accessed 2009-04-01)

[6] Bluetooth Frequently Asked Questions - http://www.mobileinfo.com/Bluetooth/FAQ.htm#t6 (Accessed 2008-12-12)

[7] iCalc – Bandwidth Calculator - http://www.ibeast.com/content/tools/band-calc.asp (Accessed 2008-12-12)

2.1.5 Mobile Handsets

The basic digital postbox must have a mobile phone handset with Bluetooth and a camera. Bluetooth is required to connect the phone to whatever device will be acting as the bridge to the higher bandwidth landline. A wireless connection to the bridge is preferable to USB as there is a risk that the USB cable might be dislodged during use resulting in the system not functioning correctly. Another consideration is none standard USB connections at the phone end. Using the USB connection would allow for the phone battery to be charged for free while the data transfer is in progress.

Similarly to the bridging device, the phones must be extremely tough and rugged to deal with the harsh conditions it will be expected to endure. A handset with an open-source operating system would also be extremely useful as it would allow easier development of the message packaging software.

SIM-card free handsets that meet the basic requirements for suitable handsets are available from independent UK high street stores for roughly £70 each.

Make	Model	Price
Sony Ericsson	T280i	£68.95
Samsung	E250	£78.95
Nokia	2630	£69.95
Nokia	2760	£69.95

FIGURE 6: A TABLE DISPLAYING THE FOUR CHEAPEST SUITABLE SIM-FREE MOBILE HANDSETS
SOURCE: CARPHONE WAREHOUSE, 399 ECCLESALL ROAD, SHEFFIELD S11 8PG

2.1.5.1 Apple iPhone

Originally released in 2007, Apple's iPhone made use of a large touch sensitive display instead of a keypad allows for developers to make much more flexible and user friendly software. Apple also released a software developer kit (SDK) for the iPhone[8] which makes the iPhone an attractive platform for this project. Unfortunately the SDK is only accessible through a $99 (US Dollars) subscription.

The iPhone is not available in the UK as a SIM free version; it costs £342.33 for pre-pay or £32.46 per month over a two year contract (O2). The huge cost of a single iPhone makes it difficult to justify using it within the context of this project.

Also of concern is the iPhone's durability. The large touch sensitive screen could be prone to heavy scratching potentially reducing the phone's usefulness over an extended period of operation.

2.1.5.2 Google Android

Developed by the Open Handset Alliance, Android is an open sourced software platform for mobile phones. The advantage of Android is that the software development kit is completely free.

[8] iPhone Developer Program – Apple Developer Connection - http://developer.apple.com/iphone/program/ (Accessed: 2009-04-30)

Unfortunately compatible devices have been slow to appear on the UK market, with only two phones available at the time of writing. These are the G1, only available on T-Mobile[9], and the G2 (also called the HTC Magic) which is only available on Vodafone[10].

Once again the high cost of Android compatible devices, £540 and £630 respectively, overshadows the potential of having a free and open development platform. Another disadvantage is the fact that both phones are only available on an 18 month contract.

2.2 Software

All the hardware in the world is useless without software to tell it what to do. The software part of the digital postbox system is especially important because it will have to interface several devices with each other as well as performing self tests on the equipment to check it is performing as it should. Without the controlling software the hardware is useless.

As with the hardware the software must meet certain criteria to be considered for inclusion in the greater system. These are as follows:

- It must be undemanding on the hardware
- Easy for the user to use, without too much initial training
- High error and fault tolerances

2.2.1 Symbian

Developed by a partnership which included Nokia, Motorola and Psion in the late 1990's, it was intended to be a platform for mobile phones. It brought together standardised libraries and user interface. Until recently Symbian was closed source propriety software with Nokia being the majority shareholder in the company. In June 2008 Nokia decided to make Symbian open-sourced[11]. They managed to achieve this extremely quickly with the final shares being purchased just six months later in early December 2008[12].

Most of the major mobile manufacturers use Symbian on their handsets (notably Nokia, Samsung and Sony Ericsson). About 64% of all mobiles shipped in 2008 used Symbian[13].

Vendor	2008 Unit Shipments	2008 Market Share
Nokia	468.4	39.7%
Samsung	196.7	16.7%
LG Electronics	100.7	8.5%

[9] Combi Phone Tariff – T-Mobile - http://www.t-mobile.co.uk/shop/mobile-phones/phones/pay-monthly/t-mobile/g1-black/combi/plans/ (Accessed: 2009-04-30)

[10] HTC Magic – The new Google Android mobile phone from Vodafone - http://shop.vodafone.co.uk/shop/mobile-phone/htc-magic/ (Accessed: 2009-04-30)

[11] Symbian | Mobile leaders to unify the Symbian software platform and set the future of mobile free - http://www.symbian.com/news/pr/2008/pr200810018.asp (Accessed: 2009-04-01)

[12] Symbian | Symbian Foundation moves closer to launch - http://www.symbian.com/news/pr/2008/pr200811007.asp (Accessed: 2009-04-01)

[13] Worldwide Mobile Phone Market Declines by 12.6% in Fourth Quarter, More Challenges To Come Says IDC - http://www.idc.com/getdoc.jsp?containerId=prUS21659209 (Accessed: 2009-04-01)

Motorola	100.1	8.5%
Sony Ericsson	96.6	8.2%
Others	218.5	18.5%
Total	1181.0	100.0%

The large market share of Symbian based mobile phones makes it an ideal developmental platform to use due to wide availability of handsets using it.

2.2.2 Linux

Linux is a free, open sourced operating system which is designed to run on a variety of platforms. It is available for free under the GNU General Public License. This makes it ideal for this project it means that multiple devices can be running the same copy without having to purchase a license for them.

The advantage of running Linux is that it is very powerful and has the capacity to run full-sized applications making it an ideal development platform. There are various distributions of Linux designed to run on small low powered devices.

2.3 Summary

In conclusion the digital postbox requires a variety of different yet inter-connectable components. While cost is one of the major factors in deciding the suitability of the components I will use in the construction and development of the project I must also take practicality and fitness of purpose into consideration.

Software wise I think a free operating system like Linux is essential to keep both the initial setup costs and the future ongoing costs down. With a free operating system there are no concerns regarding ongoing licensing fees should there be a successful deployment. Preferably a small distribution will be used to keep processing overheads to a minimum.

Using Linux as the operating system on the postbox dictates that a proper computer must be used as the bridging device. This means the only real option is one of the fan-less mini-ITX based systems. The system I propose to buy is the M61G, while it is only 50 pence cheaper than the M73G there is no purposeful gain in system performance by purchasing the more expensive system.

When it comes to a mobile phone to be given out to the local people it is quite clear that a Symbian based device has the most potential. The huge market share that Nokia and other Symbian users have gives a massive range of potential handsets to use. The cost of suitable Symbian based handsets is roughly 4.5 times cheaper than the iPhone and over 7 times cheaper than the Google G1. While those phones have easily accessible software developer kits and established developer communities, the phenomenal cost of each handset makes them prohibitively expensive to be even considered as viable for this project.

3 Development

As discussed in the last section the decision was made to use mini-ITX based hardware coupled with a Linux based operating system for the actual digital postbox due to the potential flexibility of the hardware and software.

An order was placed on the Mini-ITX website for the M61G with 512MB of RAM and a 1GB IDE flash drive. While that was on order I sourced a Bluetooth adapter from a local computer shop and set about trying various small distributions of Linux on my laptop.

3.1 Hardware Installation

Upon receiving the hardware it was checked to make sure it was what had been ordered. After that had been confirmed I was able to start putting it all together.

Firstly I removed the outer casing from the box and discovered one of the lower screws had been put in at a slight angle and this had removed the thread for that screw. However the other three screws were fine and can hold the case closed securely.

Next I got the RAM module slotted into the RAM slot towards the front of the main board. It clipped in easily with the white clips on board holding it in position.

FIGURE 7: A PHOTOGRAPH OF THE 512MB RAM MODULE

The next thing to install was the flash storage module. This had a notch on the base indicating which way round it should be inserted into the IDE connector at the very front of the board. I was careful when fitting this as I did not want to damage the pins on the board.

FIGURE 8: A PHOTOGRAPH OF THE 1GB IDE FLASH MODULE

The flash drive came with a power cable adapter which slots into the standard Molex power cable that most normal computers use. The small plug went into the top of the module. I had some issues with the half Molex at the other end. It was not easy to work which way round or indeed which two of the pins it was supposed to connect to. After some trial and error I discovered that it needed to be plugged in with the flat end attached to the red pin and the slanted end connected to the black pin. Please note in Figure 10 the power connector is not correctly inserted.

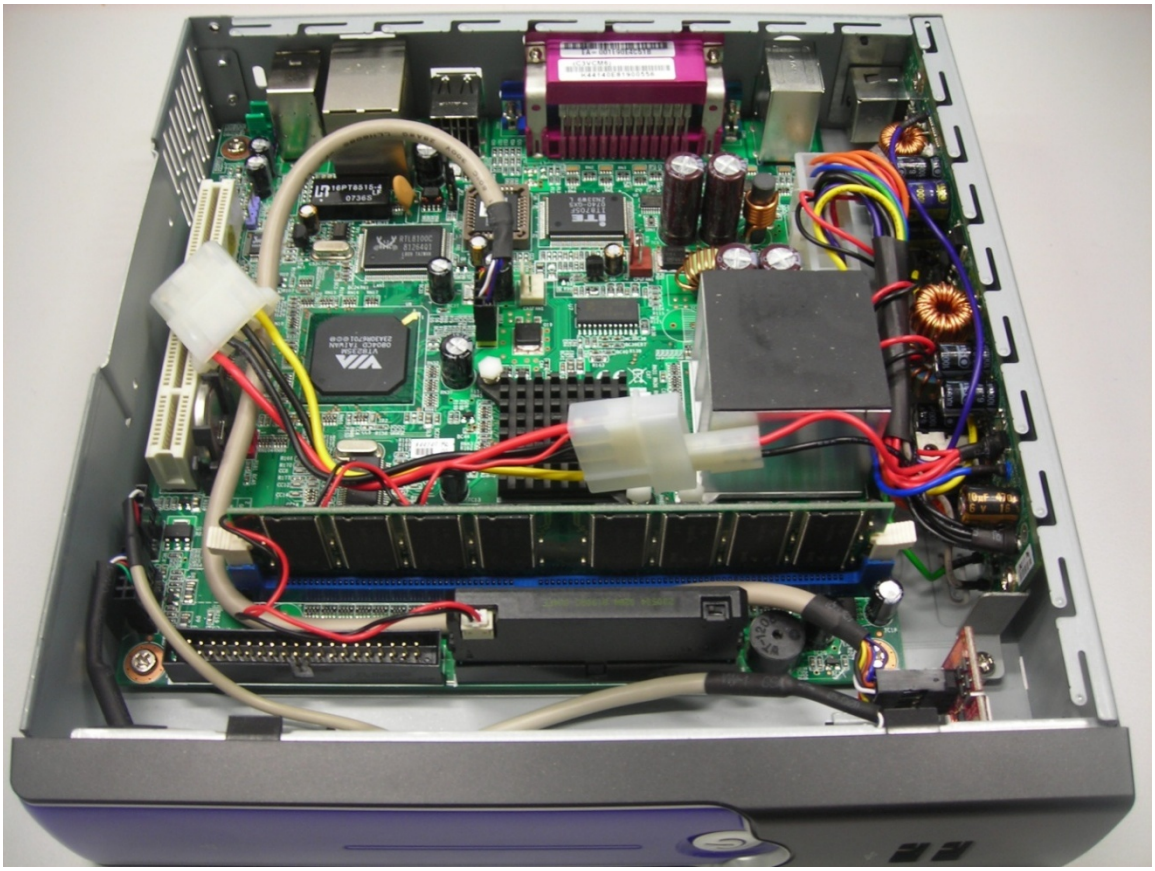

FIGURE 9: A PHOTOGRAPH OF THE MAIN BOARD WITH ALL THE COMPONENTS FITTED

After I had finished installing all the components I reattached the outer casing and slotted the box on to the stand provided.

FIGURE 10: A PHOTOGRAPH OF THE COMPLETED CASE

FIGURE 11: A PHOTOGRAPH OF THE BLUETOOTH ADAPTER

In normal operation only the power cable, the Bluetooth adapter and access to the landline, either via the Ethernet port or a USB analogue modem would be plugged in; however for ease of development a keyboard, mouse and screen were also attached.

3.2 Software Installation

To ensure a smooth installation and setup when the mini-ITX hardware arrived I decided to try various Linux distributions to check their suitability. To do this I used my laptop which is a similar specification to the final mini-ITX hardware. To simulate the limited storage space that would be encountered on the actual hardware I decided to use a 1GB SD card.

The first distribution I tried was called DamnSmallLinux[14]. This version of Linux is specifically designed to run on low end, obsolete or restricted hardware. However this distribution suffered with a lack of suitable drivers for basic hardware; for example my laptop's trackpad would not work at all, the wireless card also wouldn't work meaning I couldn't access the internet or my local network to download updates.

Another small distribution of Linux I tried is called Puppy Linux which is also designed to run on limited hardware; however Puppy Linux had slightly larger driver library and most of the functionality was returned to my laptop. I even managed to get the Bluetooth adapter working from the command line. I was able to successfully transfer various files to and from my phone using the computer however I could not get the system to accept the phone as the controlling device.

There was also a problem installing additionally software under Puppy as it did not come with a C compiler as standard. This meant installing additional software critical to the development and implementation of the system was incredibly difficult. One of the major problems I encountered was when I tried to install a C compiler for Linux called GCC so I could install Python. In order to successfully install GCC I needed a C compiler installed. This put me back to square one.

In order to get out the infinite C compiler loop I decided to use one of the most popular, full sized Linux distributions called Ubuntu which comes with most of the latest device drivers and a lot of developmental tools pre-installed.

The problem with using a full sized distribution like Ubuntu is the additional storage space required for such a large operating system. Ubuntu 8.10 weighs in at a massive 2.9GB which is nearly three times the available storage capacity of the hardware we had ordered. In order to solve the storage problem without having to order new hardware I decided to install Ubuntu on to a 4GB USB pendrive I had available. This meant I also stayed true to the hardware requirements by having solid state storage.

In order to easily identify the USB drive containing the operating system I wrapped a length of green and yellow striped electrical tape around it. Should Ubuntu be used in future models of the postbox it would be wise to order the 4GB flash module instead of the 1GB module. This puts the cost up by £109.

Once Ubuntu was installed adding additional software as required was fairly simple. I had to install Python as well as various Bluetooth tools and utilities.

3.3 Mobile Software

At the start of the development process I was given access to a Nokia N73 which was pre-installed with some software called Kheti. Kheti allowed for a series of images to be taken then for a short voice track to be recorded. It adds all these objects to a packaged file before allowing the user to transfer the package via

[14] DSL Information – http://www.damnsmalllinux.org (Accessed 2009-04-01)

email, MMS or Bluetooth. This was perfect for prototype digital postbox. However future versions of Kheti would require the Bluetooth transfer section to be rewritten to include a way of specifying an email address as the final destination of the packaged file.

Kheti is Symbian only which means in the final system only mobile devices pre-installed with Kheti will be able to use the digital postboxes. Without the additions being made to the software all potential destinations would have to be hardcoded into the digital postbox which is less than ideal.

3.4 Postbox Software

The software running on the actual digital postbox will be written in Python 2.6.1, which at the time of writing was the most up-to-date, stable release of Python. I started development for the software on the

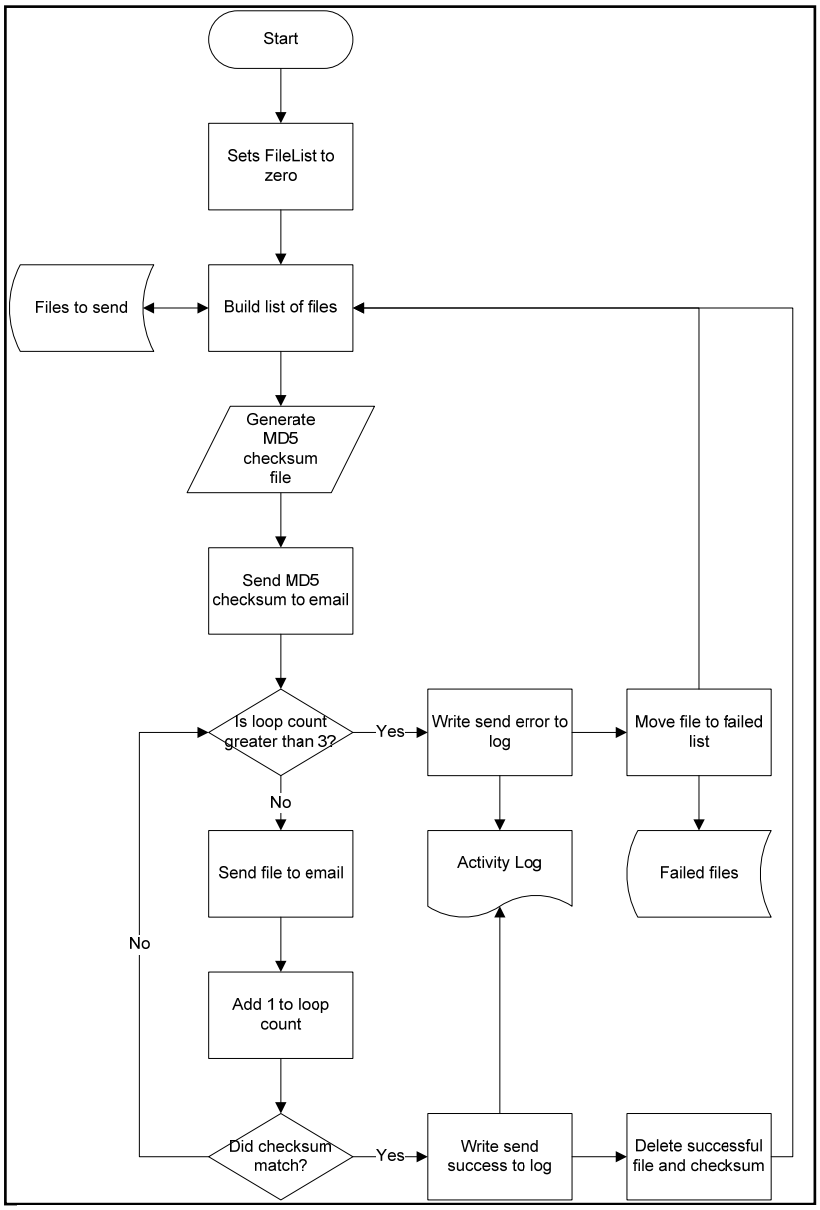

FIGURE 12: A FLOW DIAGRAM DISPLAYING THE CONCEPT OF THE SOFTWARE OPERATING ON THE PROTOTYPE DIGITAL POSTBOX

postbox on my Windows XP computer because Python programs are very easy to port to other platforms, including Linux and Symbian. All that would need to change on deployment of the DPB software would be the location of the directories.

The DPB software is designed to be nearly fully autonomous. On the initial start of the system the software will start running automatically, it will continually scan the "**to_send**" directory for new files to send.

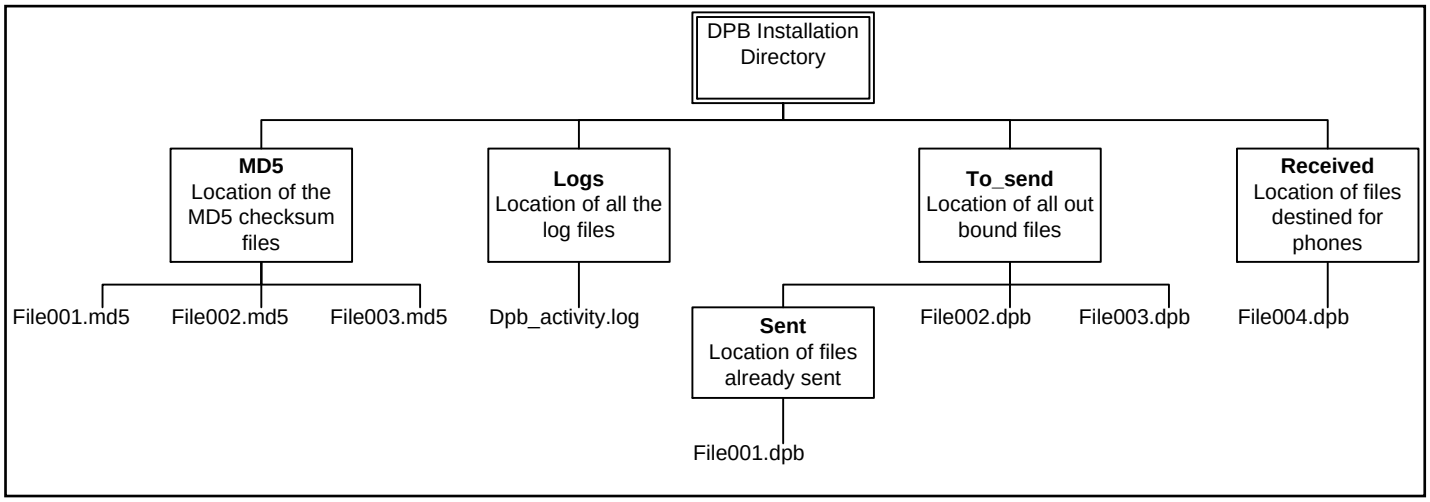

FIGURE 13: A CONCEPUTAL LAYOUT OF THE DIGITAL POSTBOX DIRECTORY STRUCTURE

When it finds a file (in this example we will follow **File001.dpb** through the system), it adds the file names to the list of files to send. It then creates an MD5 checksum for first file in the list (**File001.dpb.md5**) and saves it to the "**MD5**" folder. Checksum files that are older than 7 days are purged. The newly created checksum file is sent to the specified target address. The fact an MD5 checksum has been created and sent is written to the log file (**dpb_activity.log**).

The next thing to be done is the actual file is emailed to the target email address. Once it has been sent the file is move to the "**sent**" directory. Files are stored in the "**sent**" directory for 7 days in case of a transmission failure. The log file is updated to reflect that **File001.dpb** has been sent to the previously specified target email address.

Once every week, the system will email the log file to the administration centre; this should probably be at 01:00am on a Monday morning. If a log file is not be received within 24 hours of the estimated send time, the digital postbox that failed to send a log file will be flagged for investigation by an engineer.

4 Testing

Testing of the hardware was ongoing throughout the project. With the only issues being the lack of storage space coming from a using a much larger distribution of Linux than originally anticipated. During the building of the digital postbox system I had a lot of trouble trying to get Bluetooth to work under a Linux environment. I believe this stems from using mismatched versions of various Bluetooth protocols.

In-situ environmental testing of the hardware was not possible to replicate easily in the UK. However the postbox was left running for nearly 3 weeks straight with no major increase in case temperature or degradation in system performance.

Because of the problems encountered with Bluetooth rigorous testing of the complete system was not possible in the remaining time frame.

5 Critical Evaluation

My planning and time management for this project was initially pretty good with a lot of the research stage of the project completed before Christmas. I was unfortunately rather ill over the Christmas period and I did not manage to get the required hardware ordered until much later than I would have liked. This delay in turn reduced the amount of time I had available for development, testing and more latterly the documentation. The delay caused by the late ordering of the hardware was made worse by the fact that there was a problem with a late delivery from the hardware supplier.

While the delays were for the most part out of my hands I still should have made sure the hardware was ordered much earlier than it was. I should have also identified this as a major potential problem when planning the future workload. I also neglected to set myself milestone dates of when parts of the project should have been completed by.

Development went pretty much according to plan with the only real stumbling block being not being able to get Bluetooth to work in a Linux environment. I now believe that the problem lies with me unwittingly using protocol stacks that were not compatible with each other. The solution lies with using Ubuntu 9.04 and obexpushd 0.8 both updated programs only became available to the public after I had to cease development.

I feel I did not leave enough time to complete the documentation side of the project. Report writing is my weakest area and although I did identify this as a risk to the project I still did not factor in enough time to complete it to a standard I would have preferred.

6 References

6.1 Credits and Thanks

I wish to extend many thanks to the following people for their support, advice and help.

Dr Andrew Dearden for setting me on this crazy roller coaster of project and his mentoring over the year

My parents for their continued love and support through-out the year

Russell Bryans for his unrivalled Linux suggestions

6.2 Bibliography

Dawson, M. (2003). *Python Programming for the Absolute Beginner.* Boston: Premier Press.

Ganguli, M. (2002). *Getting Started with Bluetooth.* Rocklin: Calif.

Huang, A., & Rudolph, L. (2007). *Bluetooth Essentials for Programmers.* Cambridge: Cambridge University Press.

iCalc - Bandwidth Calculator. (n.d.). Retrieved 12 12, 2008, from http://www.ibeast.com/content/tools/band-calc.asp

IDC Worldwide Quarterly Mobile Phone Tracker. (2009, 2 4). *IDC - Press Release - prUS21659209.* Retrieved 4 1, 2009, from IDC: http://www.idc.com/getdoc.jsp?containerId=prUS21659209

Jones, M., & Marsden, G. (2006). *Mobile Interaction Design.* Chichester: Wiley.

Martelli, A. (2006). *Python In a Nutshell* (2nd Edition ed.). Sebastopol: O'Reilly Media Inc.

Muller, N. J. (2001). *Bluetooth demystified.* New York: McGraw-Hill.

O2. (n.d.). *O2 - The iPhone 3G. The iPhone You've Been Waiting For*. Retrieved April 30, 2009, from O2: http://o2.co.uk/iphone

Tarn, J. M., Pang, C., Yen, D. C., & Chen, J. (2009). Exploring the implementation and application of Bluetooth technology in the shipping industry. *Computer Standards & Interfaces* , 48-55.

7 Appendixes

7.1 Appendix 1: OpenWRT Analysis

See the OpenWRT hardware analysis spreadsheet. (New Hardware List.xls)

7.2 Appendix 2: Python Code

See the script file. (dev\dpb\dpb_main.py)

7.3 Appendix 3: List of Figures

Here is a list of all the tables, diagrams and photos used in the document.

Figure 1: Digital Postbox Concept Diagram 5

Figure 2: List of Suitable OpenWRT Compatible Hardware 6

Figure 3: A Comparison of Fan-less Mini-ITX Systems 7

Figure 4: A List of USB Powered Analogue Modems 8

Figure 5: Minimum Theoretical Transfer Speeds Via a Landline. Source: iCalc – Bandwidth Calculator - http://www.ibeast.com/content/tools/band-calc.asp (Accessed 2008-12-12) 9

Figure 6: A table displaying the four cheapest suitable SIM-free mobile handsets Source: Carphone Warehouse, 399 Ecclesall Road, Sheffield S11 8PG 10

Figure 8: A photograph of the 512MB RAM module 13

Figure 9: A photograph of the 1GB IDE flash module 13

Figure 10: A Photograph of the main board with all the components fitted 14

Figure 11: A photograph of the completed case 14

Figure 12: A photograph of the Bluetooth adapter 14

Figure 13: A flow diagram displaying the concept of the software operating on the prototype digital postbox 16

FIGURE 14: A CONCEPUTAL LAYOUT OF THE DIGITAL POSTBOX DIRECTORY STRUCTURE.....17